

## Online in-hand object localization

Maxime Chalon, Jens Reinecke, Martin Pfanne

*Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany*

*E-mails: {Maxime.Chalon, Jens.Reinecke}@dlr.de*

**Abstract**—Robotic hands are a key component of humanoids. Initially more fragile and larger than their human counterparts, the technology has evolved and the latest generation is close to the human hand in size and robustness. However, it is still disappointing to see how little robotic hands are able to do once the grasp is acquired due to the difficulty to obtain a reliable pose of the object within the palm.

This paper presents a novel method based on a particle filter used to estimate online the object pose. It is shown that the method is robust, accurate and handles many realistic scenario without hand crafted rules. It combines an efficient collision checker with a few very simple ideas, that require only a basic knowledge of the geometry of the objects. It is shown, by experiments and simulations, that the algorithm is able to deal with inaccurate finger position measurements and can integrate tactile measurements.

The method greatly enhances the performance of common manipulation operations, such as a pick and place tasks, and boosts the sensing capabilities of the robot.

### INTRODUCTION

The Institute of Robotics and Mechatronics of the German aerospace center has been developing robotic hands since more than 15 years [1]. Initially more fragile and larger than their human counterparts, the technology has evolved and the latest generation is comparable to the human hand in size and robustness (see. Fig. 1).

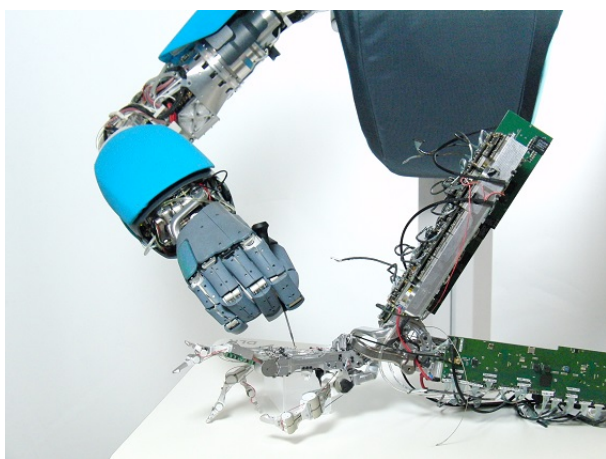


Fig. 1. Photograph of the hand of the Hand Arm System.

Motion planning and grasp planning developed at a similar speed. Most of the problems considered complex offline twenty years ago can be solved online. The problem of grasp acquisition has been well covered in simulation (e.g. GraspIt [2]) with the use of powerful computers and support

of graphics acceleration as well as in experiments with the use of impedance control.

Most robotic hands are not very dexterous once the grasp has been acquired. Even the seemingly trivial task of detecting abnormal conditions (e.g. grasp failure) is a challenge on its own. It is commonly realized by a set of hand crafted rules and therefore time consuming and error prone. Adding more objects requires more rules and can easily turn into a logic nightmare.

A recurring issue in pick and place applications with humanoid hands is the motion of the object with respect to the palm during the grasp acquisition phase. Indeed, uncertainties in the initial object pose, in the object model and in the hand control are leading to a grasp execution different than the planned version. Compliance in the finger control is used to cope with the uncertainties and makes the grasping of objects more robust. However, the equilibrium position of the object, which is passively found, cannot be controlled. Therefore, the object often moves within the palm although the grasp is eventually stable. Thus, the use of compliant grasping makes the process more robust but decreases the precision of the object position even with the most accurate hardware. The motion of the object w.r.t. the palm is not an issue on its own for the picking phase. However, the issue appears when placing the object since a tilted object may be placed in an unstable configuration and tip over. A second case for which the pose is required is the control of the pose of the object in the hand, referred to as in-hand manipulation, which requires the availability of the pose of the object w.r.t. the palm.

In most robotic systems, the pose is given or acquired from a vision system [3], [4]. This is a very effective approach as long as the occlusion of the object is minimal. However, since the objective is to estimate the pose of the grasped object, it is clear that occlusions are to be expected.

While there has been extensive research on visual localization, the problem of determining the pose of a grasped object has rarely been addressed in the literature. The use of real-time visual object tracking as part of a grasp planning system was reported in [5]. Gadenye *et al.* [6] used force controlled robots and tactile sensors to localize stationary objects. They proposed an estimation algorithm based on particle filtering. However, the great calculation effort restricted their system to three DOF scenarios. Petrovskaya *et al.* [7] build on that approach, extending it to six DOF. They implemented a particle filter that scales in precision as the localization progresses. The estimation of the pose of the

object from contact information was introduced in [8], [9]. Hypotheses for possible contact points were generated by comparing finger measurements with a previously generated model. Subsequently, these hypotheses were used to find an object pose that best represents the measurements. In [10], [11], the hand-object configuration was tracked from tactile sensing using particle filtering. The authors applied a new measurement model to account for points on the manipulator that are known not to touch the object. However, unlike this paper, the work was not considering kinematic data. The simultaneous tracking of the object and estimation of parameters of the dynamic model was proposed in [12] but the authors only applied their particle filter to pre-recorded measurements. Finally, work based on learned rules on the prediction of object motions by pushing proved to work well but was not extended to multiple contacts points [13], [14].

In this paper, a method based on a particle filter, combined with a custom collision checker is used to detect grasp acquisition faults or abnormal conditions. The algorithm is able to deal with inaccurate finger position measurements and can integrate tactile measurements.

In a first section, the problem is described and the key ideas are presented graphically on some simplified cases. The second part transforms the problem into a mathematical form with the help of the stochastic framework. The third part presents the implementation of the particle filter and details the different contributions of the collision and contact modules. It is shown that a valuable performance improvement is obtained with the use of the grasp matrix so as to obtain better predictions. The fourth part presents the simulations and experimental results obtained on the latest generation of robotic hand design at DLR. The robustness and the accuracy of the method are demonstrated with the help of realistic scenarios.

## I. PROBLEM DESCRIPTION AND KEY IDEAS

In the first part of this section, the problem of in-hand object localization is described. A second part presents the key ideas in a graphical form to help understand the mathematics needed to follow the rest of the paper.

### A. Problem descriptions

The localization of an object w.r.t. the hand is a very generic problem and the choice of a particular method is conditioned by the available sensors. This paper considers the following problem:

- 1) how to detect invalid (i.e. nearly impossible) situations?
- 2) how to estimate, **online**, a better object pose than the initial pose?

and the following assumptions are made:

- a 3D mesh of the grasped object is available.
- the 3D mesh of the hand (palm and fingers) is available.
- the position of the fingers is available online.
- an initial guess of the object pose is made (e.g. vision system).

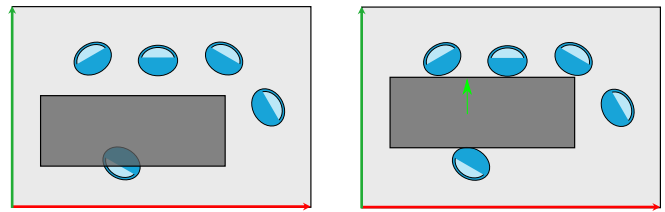


Fig. 2. Collisions are unlikely so the right hypothesis is more plausible than the left one.

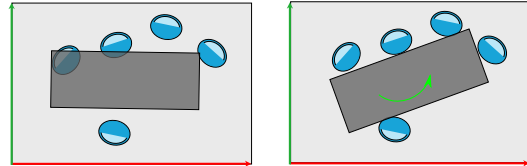


Fig. 3. Measure that allows to correct the object position by rotation.

- contact sensor information is available either as on/off or more detailed.

It is important to note that the use of visual feedback is not considered. It is required that the estimation is available online therefore, approaches that are applied offline, either at planning stage or after execution, are not in the scope.

### B. Key ideas

The algorithm developed in this paper relies upon several key ideas. It is important to keep in mind that, in real life application, the available data is not perfect, the meshes, the mass or the friction coefficients are most likely unknown. Therefore, putting effort in computing dynamic effects for an object of which the mass is known with 10% error is unlikely to lead to an appropriate answer. Similarly, it is not possible to describe all objects with analytical expressions, therefore it is necessary to use meshes. Meshes are only a discrete approximation of the object surface and introduce errors. Finally, the computation of the contact dynamics requires the knowledge of the friction coefficients and the contact type. These coefficients are very difficult to obtain since they depend on the materials, the surface, the contact geometry, and other parameters that are unlikely to be available. Therefore, the present work focuses on facts that do not require a precise knowledge of those quantities such as:

- collisions : the object can not be where the fingers are (cf. Fig. 2, 3).
- contacts : if a contact is detected the object must be triggering the sensor (cf. Fig. 4).
- grasp : without extra information the grasp matrix is a good prediction of the object motion (cf. Fig. 5).
- update : the object is immobile unless a new hypothesis is significantly improving the results

Those facts are very generic and the structure of the algorithm allows to add other facts. The presented work focuses on static facts, however more elaborate facts, such as *objects are not flying*, can be added if all required parameters are available.

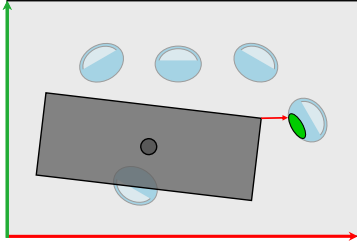


Fig. 4. In the case where contacts are detected, the object should be explaining the triggering (the green area represents an active contact).

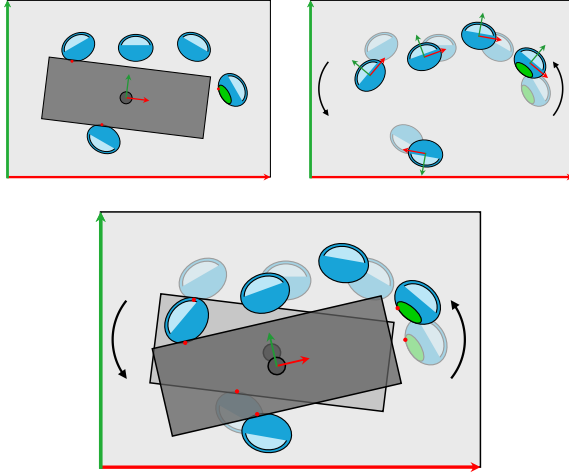


Fig. 5. Without any extra information, the object should move according to the fingers (prediction using the contact points).

## II. MATHEMATICAL FRAMEWORK

This section presents the mathematical tools that are required to understand the implementation of the localization algorithm. First, the very general and well known Bayes filter algorithm is presented. The implementation uses a particle filter because, multi-modality is required, for example, to handle distinct clusters of hypothesis. Moreover, Kalman filters (regular or extended) perform best with an analytic description of the measurement model. In a second section, the general structure of a particle filter is described. The third section describes the construction of a grasp matrix that is used for the prediction step.

### A. Bayesian filter algorithm

The Bayesian filter algorithm is the most general algorithm for calculating probabilistic state distributions [15], [16, ch.2]. The distribution of the state that is to be estimated is represented by the belief. The belief over state  $x$  at time  $t$  is denoted as  $bel(x_t)$ . At every time step the current belief is inferred from the new measurements and the control data. Starting from an initial belief  $bel(x_0)$ , the current estimate is calculated in two steps. The prediction step calculates the belief  $\bar{bel}(x_t)$  from the previous belief  $bel(x_{t-1})$  and the current control data  $u_t$ :

$$\bar{bel}(x_t) = \int p(x_t|u_t, x_{t-1})bel(x_{t-1})dx, \quad (1)$$

where  $p(x_t|u_t, x_{t-1})$  is called the state transition probability. During the update step the posterior belief  $bel(x_t)$  is calculated from the measurements that were made at time  $t$ :

$$bel(x_t) = \eta p(z_t|x_t)\bar{bel}(x_t), \quad (2)$$

where  $p(z_t|x_t)$  is the measurement probability and  $\eta$  is a normalizing constant. These two steps are executed at every time step to track the progression of the belief over the state  $x$ . The Bayes filter algorithm provides a general approach to calculate belief distributions from the measurements and the control data. However, it does not state how these distributions are represented. Therefore, there exist a number of different implementations of the Bayes filter algorithm. In this paper the particle filter is used due to its multi-modality and discrete representation of the control and measurement models.

### B. Particle Filter

The particle filter is a discrete algorithm that infers a state estimation from the control data and the measurements [17]. It represents the belief state by a set of particles  $X_t$  and their respective weights  $W_t$ :

$$X_t := x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[M]}, \quad (3)$$

$$W_t := w_t^{[1]}, w_t^{[2]}, \dots, w_t^{[M]}, \quad (4)$$

where  $x_t^{[m]}$  and  $w_t^{[m]}$  denote individual elements of the sets  $X_t$  and  $W_t$  at time  $t$ . Each particle is a concrete hypothesis of the state that is to be estimated. The set of particles is sampled to approximate the belief  $bel(x_t)$ . The evolution of the particles is calculated from control data and measurements in several steps: the prediction, the update and the re-sampling. In the prediction step control data at time step  $t$  is applied to the filter by calculating  $x_t^{[m]}$  from  $x_{t-1}^{[m]}$  and  $u_t$ . It represents sampling from the state transition probability  $\int p(x_t|u_t, x_{t-1})bel(x_{t-1})dx$  introduced in (1):

$$x_t^{[m]} \sim p(x_t|u_t, x_{t-1}^{[m]}). \quad (5)$$

In the update step the weights  $W_t$  are calculated from the measurements made at time step  $t$ . Each weight represents the probability that the measurements match the respective particle. Therefore, the set of weighted particles are a representation of the posterior belief  $bel(x_t)$ . The weights can be calculated from the measurement probability  $p(z_t|x_t^{[m]})$  used in (2):

$$w_t^{[m]} = p(z_t|x_t^{[m]}). \quad (6)$$

Using the weights that were calculated in the update step the re-sampling can be executed. In this step a new set of particles is drawn from the old set. The probability that a particle is drawn from the previous set depends on its respective weight. By re-sampling the particles according to their weights the new set represents the posterior belief  $bel(x_t)$  according to (2).

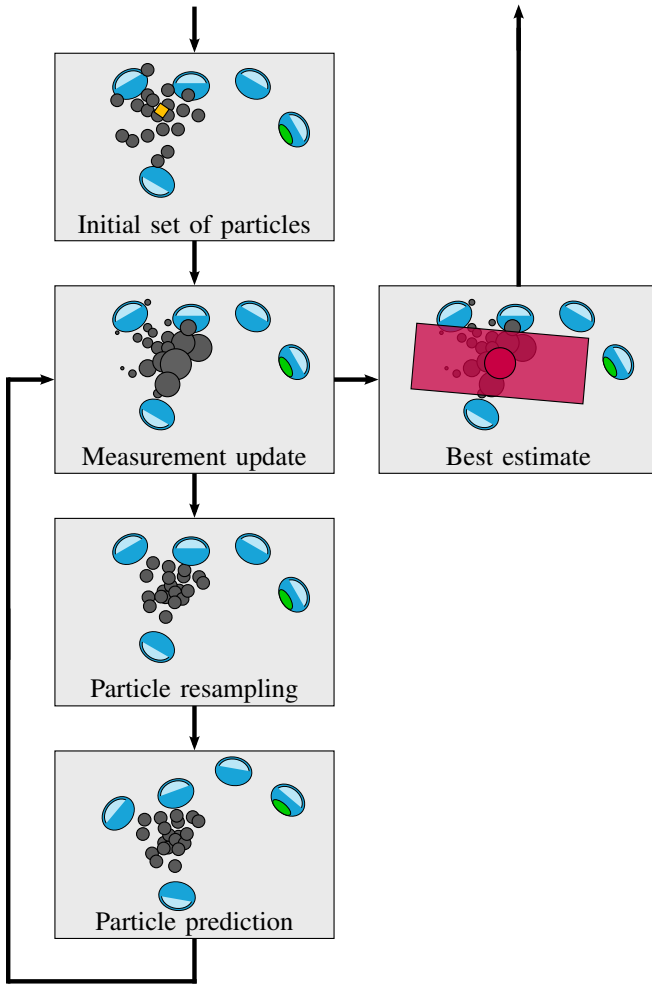


Fig. 6. Filtering workflow

### C. Grasp Matrix

The particle filter contains a prediction step that can incorporate the inputs of the actuator to adjust the current belief. In the specific case of the in-hand manipulation, the commanded motion of the fingers is a valuable knowledge in order to predict the change of pose of a grasped object [18, ch.28]. Simply put, the average motion of the contact points corresponds to the object motion. However, because the location of the contact points are not known, the first step consists in estimating them for the current belief and the detected contact points. In a second step, the contact condition can be stacked in order to yield what is known in the grasping literature as a grasp matrix. The matrix expresses, component wise, the relationship between the velocity of the contacts and velocity of the object according to a contact model. Finally, the pseudo-inverse of the grasp matrix, multiplied by the contact displacement, gives a least-square estimate of the object motion. Mathematically, the relation is

$$\dot{c}_i = \mathbf{G}_i^T \dot{x} \quad , \text{ with } i \in [1 \dots m] \quad , \quad (7)$$

where for  $m$  contacts,  $\dot{c}_i$  is the contact velocity,  $\mathbf{G}_i$  is the matrix composed of the relevant lines of the grasp matrix and  $\dot{x}$  is the velocity of the object pose. As it will be described in



Fig. 7. Graphical representation of the initial object pose by the vision system

the implementation part, the contact points required for the grasp matrix calculation are given by the collision checker and/or directly provided by the contact sensors.

## III. IMPLEMENTATION

This section presents the implementation of the steps needed for the particle processing depicted in Fig. 6.

### A. Initial set of particles

An initial particle distribution is needed to initialize the particle filtering algorithm. The distribution is based on the knowledge of the object state, that is, it is statistically distributed around the pose estimated by the vision system (cf. Fig. 7). A fixed number of sample poses are generated around the initial pose by a Gaussian sampling. If a prior variance is provided by the vision system, it can be used to obtain a well distributed set of particles.

An interesting case appears when the vision system is unable to decide whether an object is flipped or not. It is the case for example when localizing a cylindrical cup only with edge matching. In such a case, the initial distribution is performed around the two principal hypothesis of the vision system. It is one of the reasons justifying the need of a multi-modal filter. The distribution will contain both hypothesis until a measurement is able to statistically rule out one case.

### B. Update from measurements

The incorporation of the measurement data into the particle filter is realized by the weight update. It evaluates for each particle how well a given hypothesis of the object pose matches the current set of measurements. In the context of this localization method, there are two types of measurements that are considered. Kinematic data, which is provided by the joints of the hands, allows the calculation of the poses of the hand links. Since the shapes of the object and the links of the hands is available, this data can be used to identify collisions between the assumed object and the fingers. Utilizing a custom collision checker, a shortest distance or deepest penetration between each link  $j$  and the object is calculated. This distance,  $d_t^{[j]}$ , is negative, if the bodies are colliding and positive, if there is no collision. To express the probability that the assumed object pose is correct, distance  $d_t^{[j]}$  is related to a scalar weight value



$p(d_t^{[j]}):$

$$p(d_t^{[j]}) = \frac{1}{2} \left( 1 + \operatorname{erf} \left( \frac{d_t^{[j]}}{\sqrt{2}\sigma_l} \right) \right), \quad (8)$$

This function was empirically chosen to assign high weight values to positive distances and small values to negative distances. The standard deviation  $\sigma_l$  of the function can be adjusted to the inaccuracy of the measurements. The distance-weight relation is illustrated on the left side of Figure 8. Under the assumptions that the measurements of the individual link poses are independent from one another, a combined weight  $p_{t,l}^{[m]}$  for all links can be expressed:

$$p_{t,l}^{[m]} = \prod_{j=1}^{N_l} p(d_t^{[j]}), \quad (9)$$

where  $N_l$  is the number of links.

Measurements from tactile sensors are incorporated in a similar way. For each sensor  $k$  that measures a contact, the distance,  $d_t^{[k]}$ , between the assumed object and the sensor is calculated. However, the relation of the distance to a weight value  $p(d_t^{[k]})$  differs from the kinematic data:

$$p(d_t^{[j]}) = e^{-\frac{1}{2} \left( \frac{d_t^{[j]}}{\sigma_s} \right)^2}. \quad (10)$$

This function only assigns high weights to values that are close to zero, since this is the expected distance between an object and a sensor that measures a contact. The right side of Figure 8 illustrates the relation. The standard deviation  $\sigma_s$  of the distribution can be adjusted to account for the uncertainty in the pose of the sensor. A cumulative weight  $p_{t,s}^{[m]}$  for multiple sensors is expressed by the product of the individual values:

$$p_{t,s}^{[m]} = \prod_{k=1}^{N_s} p(s_t^{[k]}), \quad (11)$$

where  $N_s$  is the number of tactile sensors that measured a contact. Finally, to obtain a single scalar weight  $w_t^{[m]}$  value for a particle, the cumulative weights of both, the kinematic data and the tactile sensing information, is considered:

$$w_t^{[m]} = p_{t,l}^{[m]} p_{t,s}^{[m]}. \quad (12)$$

Calculated for every particle, the update step results in a set of  $M$  weights:

$$W_t = \{w_t^{[1]}, w_t^{[2]}, \dots, w_t^{[M]}\}. \quad (13)$$

### C. Inference of the best estimate

The objective of the in-hand localization is to provide a best estimate  $\hat{x}$  of the object pose. Therefore, a single pose has to be inferred from the set of particles. A number of different methods have been proposed in literature to accomplish this task. For this application, simply selecting the particle with the highest weight proved to be most effective, since it represents the one hypothesis of the object pose that best matches the current set of measurements. A new best estimate can be determined in every time step after the particle weights have been calculated.

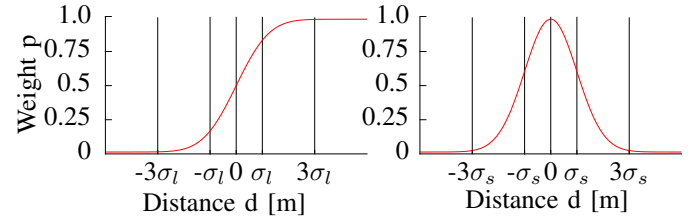


Fig. 8. Distance-weight relations for the kinematic data (left) and tactile sensing information (right). The standard deviations are denoted  $\sigma_l$  and  $\sigma_s$ .

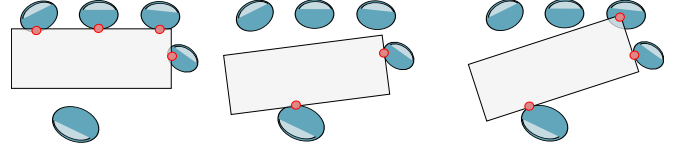


Fig. 9. Different contacts associated with the particles. The red circle are representing the contact point locations.

### D. Particle re-sampling

In the re-sampling step, a new set of particles is drawn from the old set according to their weights. In literature, a great number of re-sampling strategies have been proposed, many of which are optimized for very specific applications and constraints. To realize the re-sampling for this particle filter, the sampling importance re-sampling (SIS/R) was chosen. It is one of the most generic strategies and imposes hardly any restrictions on the belief representation. The SIS/R is executed at every time step, immediately after the calculations of the weights. The total number of particles is kept consistent.

### E. Particle prediction

Using the previous inputs on the system, such as the last commanded finger motion, it is possible to apply a *model based* transformation to each object (one for each particle). A good prediction allows to reduce the particle distribution since, otherwise, a large particle distribution required to *track* the object. For example, if a finger motion is commanded during a precision grasp, the prediction will simply move the object and everything will appear as immobile for the particles. Similarly, the particle distribution reduced if the arm motion is used to predict the object motion, instead of waiting for the particle to get collisions with the fingers. As pointed out in the previous section, the grasp matrix can be used to express this transformation. However, the actual grasp matrix is not known and it must be estimated for each particle as depicted in Fig. 9. The grasp matrix for a given particle can be obtained by defining a distance threshold for the collision checker and including the contacts measured by the sensors. Alternatively, the expected grasp matrix from the grasped planner can be used as a crude estimate, at the cost of a less realistic prediction.

It is important to note that, in general, each particle may have a different set of contact points as depicted in Fig. 9. Moreover, contacts are usually giving redundant conditions. Therefore, there usually exist only an approximate solution. A possible formulation of the problem consists in

the quadratic optimization problem defined as

$$\min_{\dot{\mathbf{x}}} (\text{stack}(\dot{\mathbf{c}}) - \mathbf{G}^T \dot{\mathbf{x}})^2, \quad (14)$$

where  $\dot{\mathbf{x}} \in \mathbb{R}^6$  is the object velocity. The stacking operator  $\text{stack}()$  creates a vertical stacking of the contact conditions for the fingers.  $\mathbf{G}^T \dot{\mathbf{x}} \in \mathbb{R}^{6 \times m}$  is the contact velocities obtained for each contact (for a given particle). One solution to this minimization problem is the Moore-Penrose pseudo-inversion that minimizes the  $L_2$  norm of the error.

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \mathbf{G}^+ [\dot{\mathbf{c}}_{i,0} \dots \dot{\mathbf{c}}_{i,j} \dots \dot{\mathbf{c}}_{i,n}]^T dt, \quad (15)$$

where  $i \in [1 \dots k]$  is the particle index ( $k$  particles),  $j \in [1 \dots m]$  is the contact point index ( $m$  contact points).  $\mathbf{x}_i^n \in \mathbb{R}^6$  and  $\mathbf{x}_i^{n+1} \in \mathbb{R}^6$  are the previous state of the object and the newly predicted one.  $\mathbf{G}^+ \in \mathbb{R}^{6 \times m}$  is the Moore-Penrose pseudo-inverse of the grasp matrix.  $[\dot{\mathbf{c}}_{i,0} \dots \dot{\mathbf{c}}_{i,j} \dots \dot{\mathbf{c}}_{i,n}]$  is the stack vector of the contact point velocities.  $dt \in \mathbb{R}$  is the time step between the last estimation and the current time (alternatively, the velocities can be replaced by the small displacements). It is important to note that the prediction step is effective as soon as it yields statistically better particle than the ones that are available. Therefore, the grasp matrix prediction can be replaced by any other prediction, such as the expected object motion provided by the planner or a few iteration of a simplified dynamic simulation.

#### IV. EXPERIMENTAL RESULTS AND SIMULATIONS

In the previous sections a particle filtering algorithm, the ideas and their implementations have been described. This section concentrates on the experimental validation of the method. In a first step the filter is evaluated with the help of simulations, which allows to analyze the influence of bias or noise in the measurements. In a second step, experiments are performed to evaluate the practical performance of the method. A representative power grasp task is demonstrating that the method allows to compensate for a tilting object. A more delicate precision grasp highlight how the estimation can be used in scenarios where using available vision tools would be extremely challenging.

##### A. Simulations

It is expected that, after some time, the particle filter is able to converge to a good estimate of the particle with maximum likely-hood. A simulation without measurement noise or bias is used to verify the execution of the algorithm. However, because the problem has several local minimums, it is not guaranteed that the best estimate is the truth although it solves properly the estimation problem. The simulation works as follow: the grasp of an object is used as initial position, an object motion is given, the finger tip position are updated using the initial grasp matrix and the object motion and the finger position are computed with the help of inverse kinematics<sup>1</sup>. The finger trajectories and the initial object pose are then used as measurements for the

<sup>1</sup>because the initial guess is good, a few Jacobian iteration yield a perfect result.

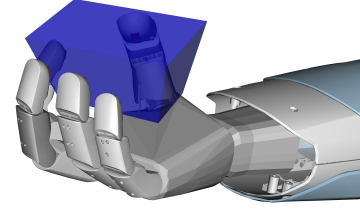


Fig. 10. Initial object pose and grasp configuration for the simulation

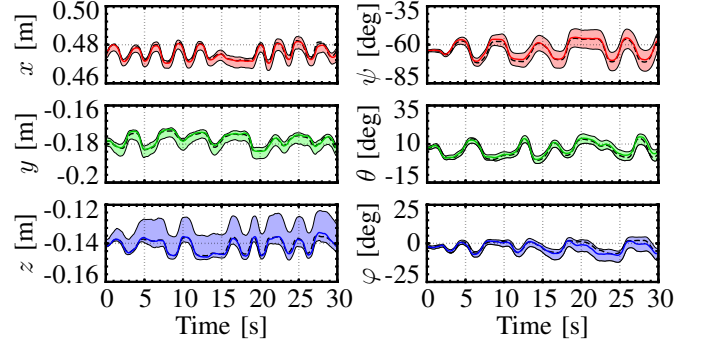


Fig. 11. Estimated object pose from ideal and inaccurate joint measurements. The reference object trajectory is depicted by the black/dashed line. The estimated trajectory from perfect measurements is represented by the colored/solid lines. The shaded area illustrates the standard deviation of the estimation from measurement with randomly added offsets to the joint positions. It highlights the effect of errors in the kinematic model of the fingers.

algorithm execution. A reference experiment with the perfect, simulated, measurement is executed in order to verify that the algorithm does not diverge. Then, multiple experiments with sensor bias and/or noise are performed to analyze the sensitivity to noise. The initial configuration of the object and the fingers is depicted in Figure 10. The estimated object pose and the reference object pose are reported in Figure 11. The solid lines are representing the results of the reference experiment, that is, the estimated pose from the perfect measurements. They show a good match with the simulated motion (represented in black/dashed). It is important to notice that the translation and the rotation around the object axis cannot be estimated and thus, tends to drift very slowly up or down. The measurable coordinates are showing an excellent tracking over a time span of 30s which ensures a good estimation of the object pose during the grasp acquisition. The drift of the non-measurable coordinates is limited to a few millimeters over the 30s and thus does not pose any problem (11, axis  $z$ ). The shaded area represents the distribution of estimation over more than 50 experiments with random initial joint position offset. The initial bias are selected according to the expected hardware accuracy and have a distribution of several degrees. Since the measurements are not anymore geometrically exact, the stochastic design takes all its importance. It is clearly visible that the values are close from the reference trajectory and that they follow very well the relative object motion. The non-measurable coordinates tend to drift faster since, in many cases, the bias in the joint position creates a squeezing effect on the object.

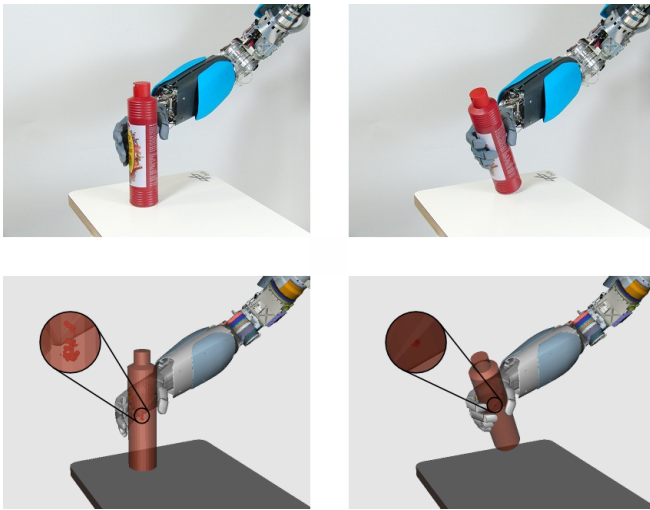


Fig. 12. Experiment: the ketchup bottle is grasped lower than expected. Left: real grasp, right: model based

### B. Experiments

According to the simulation section, the method is able to deal with uncertainties that are compatible, in scale, with the ones of the experimental platforms. The method has been applied on several robotic hands available at DLR but the results reported are obtained with the latest platform: the Hand Arm System [19]. The system is very dexterous, with 52 degrees of freedom, and anthropomorphic, thanks to its human size and force capabilities. It is the ideal platform since it can be used to manipulate common life objects, however, the method is applicable to more primitive grippers such as a two or three finger gripper. In the Hand Arm System, the finger joint angles and the contact sensor information are available from the real-time system at a rate of at least 1kHz. However, because of the calculation load involved, the particle filtering algorithm is reading them at about 20Hz. The experiment consists in grasping a ketchup bottle located by the vision system using a grasp from a grasp planner database. However, because the grasp planner did not account for the tilting during the grasp acquisition phase, the object is moving with respect to the palm. The particle filter estimates the object pose during the grasp acquisition, during the arm motion phase and the release phase. The code that handles the phases is strictly identical and no special trick is needed to make the transition between the grasped or not grasped object. Indeed, the fact that the object is attached to the end effector naturally comes from the prediction step. As a result, the algorithm is able to incorporate the pick and place concept without any prior knowledge. A more complete sequence is proposed in the media attachment that accompanies the paper, a description of it is proposed in the Appendix VI. The repeatability of the method is evaluated by executing the algorithm on the same input data several times. Because of stochastic nature of the method, the results are not expected to be identical but should be similar. In Figure 13, 20 simulations are overlaid and confirm that the method is leading to repeatable results. The results are represented along the coordinates that can be

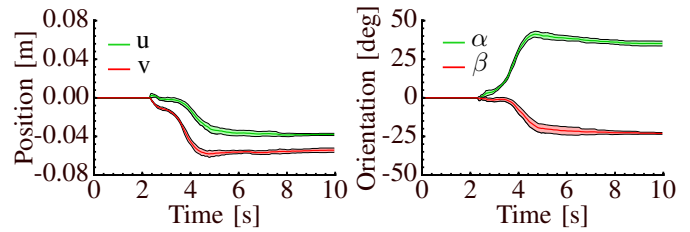


Fig. 13. Simulation: repeatability of the evaluation. The simulation is performed 20 times with the same input data. The observable coordinates  $u$  and  $v$  are reported (translations and rotations along the main axis cannot be observed).

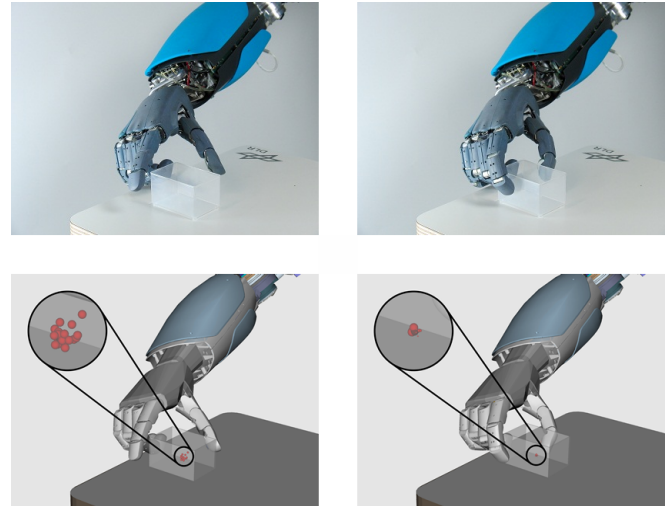


Fig. 14. Experiment: distribution of the particles during a pinch grasp of a plastic box. It is important to notice that, because of its transparency, this object is particularly challenging to track visually.

estimated. Indeed, the shape of the ketchup bottle prevents the estimation of the translation and the rotation along the vertical axis of the bottle. Figure 14 shows how the particles are gathering during a real world experiment that is very challenging with a vision system since the box is transparent. The initial particle distribution is wide since no the fingers are far from the object and are not measuring any contact. The distribution of the particle is narrowing as the fingers are getting closer. This experiment demonstrates that the method can be applied to localize the box within a few millimeters. There exists a delicate compromise between the number of particle and the computation time. Indeed, more particles increases the likely-hood of covering a better space but reduces the execution speed. Conversely, a higher speed reduces the changes between two steps and reduces the space to be covered. Figure 15 provides the relation between the number of particles and the execution time for three different simulations. It has been experimentally established that about 20 particles at 20Hz provides a good tracking and covers the space sufficiently. The execution was performed on a desktop PC CPU and a significant speed-up is expected to be achieved by using GPU computation and cluster computing since each particle weighting is independent.

### V. CONCLUSION AND FUTURE WORK

The method is based on very simple facts which, together with the efficiency of a particle filter, provide accurate,



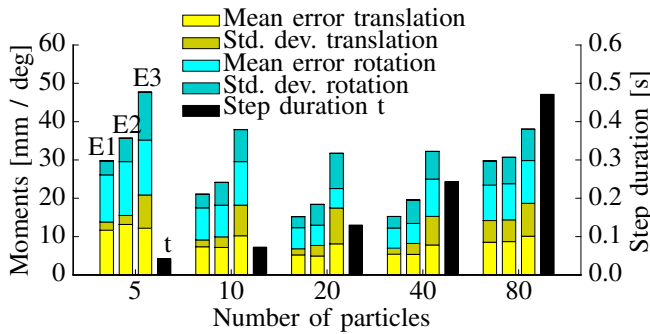


Fig. 15. Simulation: execution time vs. accuracy. E1 is a simulation with perfect initial pose and unbiased measurements, E2 uses a perfect initial pose and biased measurements (DC offset in the joint angles) and E3 uses an inaccurate initial pose and biased measurements.

robust and online pose of the grasped object. First, the key ideas were introduced and the particle filter algorithm was described. The key elements of the implementation were highlighted, such as the prediction step based on the grasp matrix. Once the algorithm was implemented, several simulations were performed to verify the stability and the repeatability of the estimation. Finally, experiments demonstrated the applicability of the approach in several realistic scenarios.

Due to its very construction, the algorithm can incorporate various facts in a flexible manner. One major next step is to add an absolute referencing system, such as vision, in order to estimate non-measurable coordinates. A second possible extension is to use the algorithm to estimate joint DC offset, thus realizing a self calibrating system.

## REFERENCES

- [1] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, "Dlr-hand ii: Next generation of a dextrous robot hand," in *Proceedings. IEEE International Conference on Robotics and Automation (ICRA)*, 2001, pp. 109–114.
- [2] A. Miller and P. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 4, pp. 110–122, 2004.
- [3] P. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 2, pp. 152–165, 1993.
- [4] M. Ulrich, C. Wiedemann, and C. Steger, "Cad-based recognition of 3d objects in monocular images," in *Proceedings. IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 1191–1198.
- [5] D. Kragic, A. Miller, and P. Allen, "Real-time tracking meets online grasp planning," in *Proceedings. IEEE International Conference on Robotics and Automation (ICRA)*, 2001, pp. 2460–2465.
- [6] K. Gadeyne and H. Bruyninckx, "Markov techniques for object localization with force-controlled robots," in *Proceedings. 10th International Conference on Advanced Robotics (ICAR)*, 2001, pp. 91–96.
- [7] A. Petrovskaya, O. Khatib, S. Thrun, and A. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," in *Proceedings. IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 707–714.
- [8] S. Haidacher, "Contact point and object position from force/torque and position sensors for grasps with a dextrous robotic hand," Ph.D. dissertation, Technische Universität München, Universitätsbibliothek, 2004.
- [9] S. Haidacher and G. Hirzinger, "Estimating finger contact location and object pose from contact measurements in 3d grasping," in *Proceedings. IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, IEEE, 2003, pp. 1805–1810.

- [10] C. Corcoran and R. Platt, "A measurement model for tracking hand-object state during dextrous manipulation," in *Proceedings. IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 4302–4308.
- [11] R. Platt, F. Permenter, and J. Pfeiffer, "Inferring hand-object configuration directly from tactile data," in *Electronically published proceeding of the Mobile Manipulation Workshop, ICRA*, 2010.
- [12] L. Zhang, S. Lyu, and J. Trinkle, "A dynamic bayesian approach to real-time estimation and filtering in grasp acquisition," 2012.
- [13] M. Kopicki, R. Stolkin, S. Zurek, T. Morwald, and J. Wyatt, "Predicting workpiece motions under pushing manipulations using the principle of minimum energy," in *Robotics: Science and Systems conference, Proceedings of the Workshop: Representations for object grasping and manipulation in single and dual arm tasks*, 2010.
- [14] —, "Learning to predict how rigid objects behave under simple manipulation," in *Proceedings. IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, IEEE, 2011, pp. 5722 – 5729.
- [15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005, ch. 2. Recursive State Estimation, pp. 13–38.
- [16] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 24–33, 2003.
- [17] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [18] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer, 2008, ch. 28. Grasping, pp. 671–700.
- [19] M. Grebenstein, M. Chalon, W. Friedel, S. Haddadin, T. Wimbeck, G. Hirzinger, and R. Siegwart, "The hand of the dlr hand arm system: Designed for interaction," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1531–1555, 2012.

## VI. VIDEO ATTACHMENT DESCRIPTION

The video shows how the method can be used in a realistic setup, all videos are in realtime and synchronous. The first part serves as a motivation example. The robot picks up the bottle and the software *attaches* the bottle to the palm. The planner then compute the release configuration according to the initial grasp configuration and the planner provides a collision free path. Finally, the robots executes the motion and releases the object. It is clearly visible that the fingers are going through the object because the object pose is not updated during the grasp. As a result the object is placed in an unstable configuration and falls. In the second part, the particle filter is used to estimate the object pose. The available knowledge is limited to the 3D models of the robot and the object, the initial position of the object and the configuration of the robot online. In a first step the particle filter is initialized with the video data and the algorithm is started **before** the motion is started. It can be seen that the fingers are not going through the object anymore and that the estimated object pose is realistically estimated. In the second phase, the planner used the estimated pose after lift off in order to plan the place configuration. The motion is executed and, because the release configuration is matching the real object pose, the object is released in a stable configuration. It is very important to note that no knowledge of a the effect of a grasp is programmed. The method, very naturally, results in the attached/detach behavior without ever being coded.